

Methoden und Werkzeuge für die frühen Phasen der Software-Entwicklung

Astrid Beck, Jürgen Ziegler

Fraunhofer-Institut für Arbeitswirtschaft und Organisation (IAO), Stuttgart

Zusammenfassung

Den zunehmenden Ansprüchen und Möglichkeiten der interaktiven Verarbeitung komplexer und umfangreicher Information steht nach wie vor ein Mangel an arbeits- und benutzerorientierten Analyse- und Gestaltungsverfahren gegenüber. Dieser Beitrag gibt einen Überblick über Methoden und Werkzeuge, die bisher in den frühen Phasen der Software-Entwicklung eingesetzt werden. Es werden darüberhinaus Ansätze anderer Disziplinen untersucht, die im Software-Entwicklungsprozeß bisher noch wenig Beachtung gefunden haben. Aufgrund nach wie vor bestehender Probleme in der Software-Entwicklung und der Defizite vorhandener Methoden und Werkzeuge werden Anforderungen an eine aufgaben- und benutzer-orientierte Methodik formuliert und ein in der Entwicklung befindliches Verfahren skizziert.

1 Probleme in den frühen Phasen der Software-Entwicklung

In den frühen Phasen der Software-Entwicklung, also während der Anforderungsanalyse, der Aufgabenanalyse und der Spezifikation des zu entwickelnden Software-Systems stellt sich nach wie vor das Problem, eine für alle Beteiligten adäquate Anforderungsdefinition zu erstellen. Durch sich häufig ändernde Anforderungen, nur unzulänglichem Wissen von Arbeitsabläufen und fehlender Benutzerorientierung wird es für den Software-Entwickler immer schwieriger, eine vollständige, aufgabengerechte und für die Gestaltung unmittelbar umsetzbare Anforderungsdefinition zu entwickeln.

Die Folge sind unzulängliche Software-Produkte, die die Benutzer jedoch nicht mehr so einfach und widerspruchsfrei hinnehmen. Es ist zu erwarten, daß Mitarbeiter in Zukunft in noch stärkerem Maße die Befriedigung ihrer Bedürfnisse und Interessen am Arbeitsplatz fordern und aktiv bei der Analyse und Gestaltung mitarbeiten wollen (Schubert, Zink 1990). Nach wie vor gibt es aber einen Mangel an konzeptueller und methodischer Unterstützung für die Beteiligung von Benutzern an Systemanalyse und -entwurf. Es ergeben sich Mißverständnisse, die möglicherweise erst sehr spät - wenn überhaupt - geklärt werden können. Wir glauben, daß dies vor allem auf das Fehlen geeigneter Kommunikationsmittel zur wirksamen Benutzerbeteiligung zurückzuführen ist.

Ein weiteres Problem liegt in der fehlenden Qualifikation aller am Prozeß Beteiligten. Auf Seiten der Benutzer fehlt meist das Grundwissen über Möglichkeiten und Probleme des EDV-Einsatzes. Auf Seiten der Software-Entwickler fehlt die Kompetenz, gemeinsam Arbeitsprozesse mit den Benutzern zu gestalten.

Die vorgefundene Arbeitssituation wird oftmals unkritisiert in den Entwurf übernommen und als Grundlage für das System festgelegt. Eine für den Benutzer ungünstige Aufgabenverteilung läßt sich später aber kaum noch korrigieren. Oft deckt sich die ermittelte Systemfunktionalität gar nicht mit der Struktur der Aufgabe und den Aufgabenzielen, oder sie ist nicht auf die Reihenfolge der Arbeitsschritte abgestimmt.

Bisher gibt es keinen systematischen Übergang von der Anforderungs- und Aufgabenanalyse zum Systementwurf. Dies gilt insbesondere für die Gestaltung von Benutzungsschnittstellen. Eine wesentliche Ursache dafür sind die Verwendung nicht aufeinander abgestimmter Methoden. Es gibt bislang keine durchgängige Methode, die den Übergang von der Analyse zur Gestaltung unterstützt.

2 Methoden zur Aufgabenanalyse

In dem vom BMFT geförderten Vorhaben TASK (Technik der aufgaben- und benutzerangemessenen Software-Konstruktion) wurden daher eine Reihe von Methoden untersucht, mit dem Ziel, Analysemethoden zu ermitteln, die den Aufgabenanalyseprozeß besser unterstützen und Ergebnisse und Ansätze für eine aufgaben- und benutzerangemessene Gestaltung von Software liefern (Beck, Müller-Haffner 1989).

Methoden zur Analyse von Bürotätigkeiten

Ziel von Methoden zur Analyse von Bürotätigkeiten ist eine problem- und zeitgerechte Gewinnung und Handlungsumsetzung von Informationen (VDI 5003, S. 5). Die meisten Methoden zur Analyse und Gestaltung von Arbeitssystemen im Büro (Schönecker und Nippa 1987) beschränken sich ausschließlich auf die Identifizierung von technischem Unterstützungsbedarf, Betrachtungen der Gesamtunternehmung und dem Kommunikationsfluß zwischen den einzelnen betrieblichen Stellen. Die Methoden haben Rationalisierung zum Ziel und orientieren sich daher vornehmlich am Interesse nach effektiver und ökonomischer Wirksamkeit. Diese Methoden blenden eine benutzerorientierte Perspektive dabei fast vollständig aus. Die psychosoziale Dimension des Analyse- und Gestaltungsfeldes wird ausgeklammert (Berr, Feuerstein und Rödiger 1989). Einige Methoden werden nur von externen Beratern angewendet oder es sind umfangreiche Hard- und Softwareanschaffungen zur Durchführung der Analyse notwendig.

Methoden der psychologischen Arbeitsanalyse

Die psychologische Arbeitsanalyse wird bei Frei (1981) definiert als die "Analyse des Prozesses, der psychologischen Struktur und Regulation menschlicher Arbeitstätigkeiten im Zusammenhang mit ihren Bedingungen und Auswirkungen." (S. 12). Das Verfahren zur Analyse von Regulationserfordernissen in der Arbeitstätigkeit (VERA, Volpert et al. 1983) und das Tätigkeitsbewertungssystem (TBS, Rudolph et al. 1987) ermöglichen beispielsweise die Analyse insbesondere von geistiger Beanspruchung. Methoden, die eine aufgaben- und benutzerorientierte Analyse auch für die Gestaltung von Software anstreben sind VERA/B (Rödiger 1987) und die Kontrastive Analyse (Dunckel 1989). Es wird die Interaktion zwischen Mensch und Arbeit untersucht, um Lösungen für unterschiedliche Problemstellungen ableiten zu können. Psychologische Arbeitsanalysen können im Hinblick auf Maßnahmen der Arbeitsgestaltung durchgeführt werden, mit dem Ziel, Handlungs- und Entscheidungsspielräume zu erhalten und zu erhöhen.

Methoden zur Aufgabenanalyse aus dem Software Engineering

Eine Reihe von Software-Engineering-Methoden wurden bereits in den 70er Jahren als Analyse- und Entwurfsmittel entwickelt (wie SADT, Petri-Netze, Entscheidungstabellen). In letzter Zeit - obwohl auch keine neue Methode - gewinnt Structured Analysis (DeMarco 1978) zunehmende Bedeutung als quasi-Standard-Methode in Systemanalyse und -Entwurf. In Structured Analysis werden Entity-Relationship- (Chen 1976) und Data-Flow-Diagramme für Analyse und Entwurf eines Funktions- und Datenmodells eingesetzt. Diese Reduktion der Arbeitsaufgabe auf den informationsverarbeitenden Aspekt läßt aufgaben- und benutzerorientierte Herangehensweisen und die Gestaltung von Arbeitsabläufen jedoch zu kurz kommen.

Methoden zur kognitiven Aufgabenanalyse

Ziel der kognitiven Aufgabenanalyse ist es, die mentale Repräsentation von Aufgaben durch den Benutzer zu modellieren, um daraus Aussagen über die Erlernbarkeit von Rechnerfunktionen, deren Konsistenz, Gedächtnisbelastungen und mögliche Fehlerquellen zu gewinnen (Ziegler 1988). Task-Action-Grammatiken (Green, Schiele, Payne 1989), GOMS (Card, Moran und Newell 1983) und Cognitive Complexity Theory (CCT, Polson 1987) sind Ansätze, die sich unter den kognitiven Methoden durchgesetzt haben. Diese Methoden beziehen sich auf eine sehr niedrige Ebene der Handlungsregulation, die im wesentlichen auf die Ebene der Ausführung abzielt (beispielsweise das Drücken von Tasten). Dabei setzen die kognitiven Methoden eine bereits getroffene Mensch-Maschine-Funktionsteilung voraus; die funktionale Gestaltung des Systems wird also nicht hinterfragt bzw. als optimal eingestuft.

Methoden zur Aufgabenanalyse aus der "Künstlichen Intelligenz"

Die KI-Forschung entwickelte ihren eigenen Weg zur Datenerhebung, da herkömmliche Software-Engineering-Methoden für KI-Probleme als ungeeignet angesehen wurden (Partridge 1989). Dies führte zu einer eigenen Variante der Softwaretechnik, dem Knowledge Engineering. Einen umfassenden Überblick zu Methoden und Werkzeugen des Knowledge Engineering gibt Boose (1989). Zum Einsatz kommen eine Fülle sozialwissenschaftlicher Methoden, "aus denen der Informatiker sich bedienen kann, die er jedoch ohne sozialwissenschaftliche Ausbildung nicht sicher und kompetent beherrschen kann" (Bonsiepen, Coy 1990). Gefordert sind also auch hier zusätzliche Kompetenzen des Informatikers beim Einsatz von entsprechenden Methoden.

Nicht nur zur Erhebung von Wissen, sondern auch zu dessen Darstellung wurden im Bereich der Künstlichen Intelligenz eine Reihe von Repräsentationsmethoden entwickelt (vgl. Freksa 1989). Repräsentationsmodelle wie Frames (Schemata, Skripts) lassen sich z.B. zur Beschreibung von Aufgaben verwenden. Das Knowledge Engineering kann also durchaus verwertbare Ansätze bieten, von Kritikern wird vor einer eigenen Disziplin jedoch gewarnt, da sie den Problemen des Software Engineering nicht begegnet, im Gegenteil diese sogar noch verstärkt (Partridge 1989). Es ist daher eine stärker umfassende Betrachtung gefordert, formale Beschreibungen allein verkennen den kommunikativen und dynamischen Charakter der zu analysierenden Arbeitssituation.

3 CASE-Werkzeuge

CASE (Computer Aided Software Engineering) - Werkzeuge sollen den Software-Entwicklungsprozeß in allen Phasen unterstützen. Die Verbreitung von CASE-Werkzeugen findet in letzter Zeit einen deutlichen Zuwachs (Rock-Evans 1989). Die Hoffnung kam auf, einerseits die steigende Komplexität der Systeme und andererseits die Probleme in der Anforderungs- und Aufgabenanalyse mit einem Software-Werkzeug eher überwinden zu können. Die oftmals sehr hochgesteckten Erwartungen wurden jedoch bis jetzt nicht erfüllt.

Betrachtet man die Werkzeuge bezüglich ihrer Unterstützung des Analyseprozesses, so nehmen alle für sich in Anspruch, Analyse zu unterstützen. Diese Unterstützung beschränkt sich jedoch darauf, "intelligente" graphische Editoren für die einschlägigen Methoden zur Verfügung zu stellen (vgl. Abb. 1). Unter dem Gesichtspunkt einer aufgabenorientierten Systemanalyse finden arbeitspsychologische oder kognitive Aspekte aber keinerlei Berücksichtigung. Konzepte, wie CASE-Werkzeuge im Team bei der kooperativen Software-Entwicklung einzusetzen sind, gibt es bislang nicht.

CASE Produkt	E/R Diagrams	DFD Diagrams	Data Dictionary	Real-time Modeling	weitere Methoden
Anatool / Blues	-	*	*	-	SSADM, LSDM SADT
Auto-Mate Plus	*	*	*	-	
Design / IDEF	-	-	*	-	
DesignAid	*	*	*	*	Structure Charts, .. Jackson
Excelerator	*	*	*	-	
Foundation (Design/1)	*	*	*	-	
Jackson-Tool	-	-	-	*	Structure Charts, .. State-Transition (ST)
IEW	*	*	*	*	
MacAnalyst	*	*	*	*	
Mac Bubbles	-	*	*	*	SEtec, SSADM, LSDM Petri-Netze
Maestro	*	*	*	*	
Pace	-	-	-	*	
Power Tools	*	*	*	*	ST, Struct. Charts ISOTEC
Predict Case	*	-	*	-	
ProMod	-	*	*	*	
Software through Pictures	*	*	*	*	ST, DT, Struct. Charts ST, Decision Table
Teamwork	*	*	*	*	
TurboCASE	*	*	*	*	
Visible Analyst Workbench	-	*	*	*	ST SSADM
VSF	*	*	*	*	

Abb. 1 CASE-Werkzeuge und Analyse-Unterstützung

Trotz der Defizite von CASE-Werkzeuge hat die Diskussion um das Für und Wider ihres Einsatzes die Aufmerksamkeit verstärkt auf die frühen Phasen der SW-Entwicklung gelenkt. Dies beweist auch das gestiegene Interesse an Methoden wie Strukturierter Analyse und Entity-Relationship-Modellierung.

4 Anforderungen an eine aufgaben- und benutzerorientierte Analysemethodik

Trotz einer Vielzahl von zur Verfügung stehenden Methoden und Werkzeugen sind die eingangs geschilderten Probleme nach wie vor akut. Der folgende Katalog von Anforderungen sollte daher von einer Analysemethodik erfüllt werden, wenn sie dem Ziel näherkommen soll, den Menschen mit seinen Arbeitsaufgaben als zentralen Maßstab für den Analyse- und Gestaltungsprozeß zu begreifen.

Integrierte, systematische Vorgehensweise

Diese Forderung zielt auf eine integrierte Analysemethodik, die technische, organisatorische und soziale Aspekte mit aufeinander abgestimmten Methoden und Werkzeugen gleichermaßen berücksichtigt. Nach Möglichkeit sind erprobte und bewährte Verfahren zu integrieren.

Flexible Methodik

Die Methodik muß so flexibel sein, daß sie den jeweiligen Gegebenheiten angepaßt werden kann und möglichst universell einsetzbar ist. Sie muß verschiedene Perspektiven erlauben, muß aber selbst einfach zu nutzen sein. In der Methodik selbst sollten Alternativen in der Vorgehensweise angeboten werden, mit dem Ziel, dem Software-Entwickler Hilfestellung bei der Auswahl von Werkzeugen und Methoden zu geben. Es sollte möglichst keine aufwendige Hard- und Software erforderlich sein.

Ganzheitliche Herangehensweise

Die Analyse darf nicht reduziert werden auf den informationsverarbeitenden Aspekt. Die Analyse und Gestaltung von Aufgaben und Arbeitsabläufen sollte vorrangiges Ziel sein. Die Einbettung der einzelnen Aufgaben in den Kontext der gesamten vorgefundenen oder geplanten Arbeitstätigkeit muß berücksichtigt werden.

Gestaltungsrelevante Merkmale erfassen

Es sollen nur die für die spätere Gestaltung relevanten Merkmale erfaßt werden, d.h. eine Methodik soll nicht mehr als nötig erfassen. Es sollten qualitative oder quantitative Beobachtungen bzw. Abschätzungen vorliegen.

Übergang von Analyse zur Gestaltung unterstützen

Die verwendeten Methoden sollen aufeinander abgestimmt werden. Es muß deutlich werden, wie die Ergebnisse der Analyse in den Entwurf umgesetzt werden sollen. Sinnvoll ist u.U. ein systematischer Übergang zum Prototyping.

Qualität

Der Begriff der Qualität generell muß ausgedehnt werden auf aufgaben- und benutzergerechte Gestaltung und bereits in der Analyse berücksichtigt werden. Unterstützt werden sollen aber auch Aspekte der Qualitätskontrolle. Dazu gehören Hilfsmittel für Projektmanagement und Produkt- bzw. Fortschrittskontrolle, Versionsverwaltung und Wiederverwendbarkeit.

Benutzerorientierung durch Partizipation

Die Interessen der Systemnutzer sollen bereits in der Analysephase durch die Möglichkeit der aktiven Beteiligung berücksichtigt werden. In diesem Zusammenhang muß eine Methodik berücksichtigen, wann und in welcher Weise die Systemnutzer einzubeziehen sind und geeignete Kommunikationsmittel bereitstellen.

Aufgabenmerkmale werden charakterisiert durch die Aspekte *Organisation, Arbeitsumgebung, Arbeitsmittel, Information*. Im Mittelpunkt des Untersuchungsinteresses steht der *Mensch* mit seinen *Arbeitsaufgaben*. Die Merkmale bezüglich der Arbeitsaufgabe und der zu verarbeitenden Information gibt die Abb. 2 wieder. Merkmale bezüglich der Arbeitsaufgaben sollten möglichst objektiv feststellbar und unabhängig von der jeweiligen Problemstellung erfassbar sein. Mit einem psychologischen Arbeitsanalyseverfahren wie VERA lassen sich die Merkmale größtenteils erfassen und bewerten und mit erweiterten Entity-Relationship- und Data-Flow-Diagrammen graphisch darstellen. Dem Software-Entwickler stehen somit eine Reihe von Kriterien zur Verfügung, die ihn hin zu einer mehr aufgaben- und benutzerorientierten Vorgehensweise führen. Die Beschreibung von Aufgaben auf der Basis von Aufgabenmerkmalen läßt sich rechnergestützt verwalten.

Es wird eine Methodik entwickelt, die die untersuchten Methoden sowie Anforderungen an eine aufgaben- und benutzerangemessene Vorgehensweise für die Analyse berücksichtigt (Beck, Ilg 1990). Im Rahmen der Analyse der Informationsflüsse werden alle für die Gestaltung relevanten Aufgabenmerkmale im Problemmodell erfaßt (vgl. Abb. 2).

6 Ausblick

Aufbauend auf dem Problemmodell wird das Lösungsmodell entwickelt. Dies beinhaltet eine Kritik des Problemmodells und den Entwurf des Funktions- und Datenmodells. Auf diese werden wiederum Sichten definiert, die die Entwicklung der Dialogschnittstelle unterstützen.

Auf Basis dieses Aufgabenbeschreibungsmodells werden Software-Werkzeuge für verschiedene Hardware (Workstation, PC, Apple Macintosh) erarbeitet. Diese unterstützen nicht nur eine konsistente und vollständige Dokumentation, sondern auch die Wiederverwendbarkeit von Aufgabenbeschreibungen für alle folgenden Phasen und für weitere Projekte. Ein Begriffsglossar enthält für Benutzer und Entwickler relevante Definitionen und Kontextinformationen. Diese stehen für alle Phasen online zur Verfügung und sollen der gemeinsamen Verständigung zwischen Benutzer und Entwickler dienen. Es umfaßt alle wichtigen Vorgänge und orientiert sich an der Begriffswelt des Anwenders.

Die skizzierte Analysemethodik und das zugrundeliegende Modell werden im Augenblick in praktischen Untersuchungen erprobt. Sind diese Untersuchungen weitgehend abgeschlossen, wird die Vorgehensweise hinsichtlich des Entwurfs erweitert. Erste Reaktionen von Software-Entwicklern und Benutzern auf die Methodik waren durchaus positiv.

Verständliche Darstellungsmittel

Die Darstellungsmittel zur Modellierung von Aufgabenmerkmalen sollen als verständliches Kommunikationsmittel für alle Beteiligten nutzbar sein. Es muß eine verständliche Repräsentation mit leicht nachvollziehbarem Bezug zur realen Arbeitssituation haben.

Kritik und Neugestaltung bestehender Arbeitsabläufe

Arbeitsabläufe sind nicht statisch aufzufassen sondern als gestaltbar zu begreifen. Eine differentielle Neugestaltung ist Aufgabe des Software-Entwicklers unter aktiver Beteiligung der Anwender. Die Gestaltung soll sich an Kriterien, wie Vermeidung von Belastungen, Erhaltung und Schaffung von Handlungsspielräumen und Förderung der Persönlichkeit orientieren. Die Analyse muß Ansatzpunkte für Bewertungsverfahren bieten.

Qualifizierung

Anwendern und Benutzern müssen Grundlagen zum Verständnis der Möglichkeiten und Einschränkungen der Software-Gestaltung vermittelt werden. Umgekehrt müssen die Systemgestalter ein Grundwissen an arbeitswissenschaftlichen und software-ergonomischen Kenntnissen erwerben, damit arbeitsorganisatorische Gestaltungsmaßnahmen getroffen werden können und eine interdisziplinäre Zusammenarbeit machbar wird.

Keine übermäßige Papierarbeit ...

Methoden und Werkzeuge müssen die Zahl der Dokumente überschaubar halten und deren Erstellung und Verwaltung unterstützen. Der Aufwand, Dokumente zu aktualisieren, ist so gering wie möglich zu halten.

5 Ein Modell zur Analyse von Aufgabenmerkmalen

In dem vom BMFT geförderten Vorhaben TASK wird in Zusammenarbeit mit zwei Software-Häusern an einer Analyse- und Gestaltungsmethodik gearbeitet, die diesen Anforderungen genügen soll. Das Ziel dieses Projekts ist die exemplarische Realisierung einer Software-Produktions-Umgebung. Unter Produktions-Umgebung wird sowohl ein CASE-Tool zur Unterstützung des Entwurfs aufgaben- und benutzerorientierter Software als auch ein Projektabwicklungsmodell und Schulungskonzept verstanden.

Zunächst wurde ein Modell entwickelt, das die Merkmale charakterisiert, die in einer Aufgabenanalyse zu ermitteln sind. Es geht davon aus, daß sich die Arbeitssituation im wesentlichen durch Aufgabenmerkmale beschreiben läßt.

Arbeitsaufgaben

Name	• Bezeichnung der Aufgabe
Art	• Aufgabentyp
Struktur	<ul style="list-style-type: none"> • Hierarchie der Teilaufgaben • Anfangszustand • Vorgänger der Aufgabe • Ablaufstruktur (Kontrollfluß) • Endzustand (Ziel) • Nachfolger der Aufgabe
Ereignis	• Auslöser für Aufgabe
Vor- und Nachbedingung	<ul style="list-style-type: none"> • was muß vor Ausführung erfüllt sein • was gilt nach Ausführung
Häufigkeit Wiederholungsrate Priorität Dauer	<ul style="list-style-type: none"> • wie oft tritt Aufgabe auf • wie oft wird die Aufgabe wiederholt • Wichtigkeit (gekoppelt an auslösendes Ereignis) • mittlere Ausführungsdauer
Vollständigkeit	<ul style="list-style-type: none"> • sequentiell • hierarchisch
Eingriffs- und Auswahlmöglichkeiten	<ul style="list-style-type: none"> • Wahl unterschiedlicher Arbeitsmittel • Wahl unterschiedlicher Verfahren • Variation in der Abfolge • Unterbrechung / Abbruch von Prozessen • Rücknahme/Stornierung
Belastungsfaktoren / Restriktionen	<ul style="list-style-type: none"> • Zeit- und Te Fehler und mögliche Konsequenzen • Fehlerrisiko • Unterbrechungen
Technische Durchführbarkeit	• Ausführbarkeit der (Teil-)Aufgabe mit Dialogsystemen -> Funktionsteilung
Sicherheit	<ul style="list-style-type: none"> • kein Verlust von Daten • kein unerlaubter Zugriff
Komplexität	z.B. <ul style="list-style-type: none"> • Zahl der Teilaufgaben • Zahl der Operationen • Zahl der Auswahlmöglichkeiten • Zahl der zu verarbeitenden Dokumente • Zahl der Entscheidungsmöglichkeiten • Vollständigkeit

Information

Struktur	<ul style="list-style-type: none"> • Name • Beschreibung • Wertebereich • Medium • Identifikation • Komponenten • Typ • Anzahl (durchschn., max.)
Relationen	<ul style="list-style-type: none"> • Relation zu Ziel • wird erzeugt von {Aufgabe(n) Entität} • wird {benutzt gelesen geändert gelöscht} von {Aufgabe(n)} • wird {benutzt gelesen geändert gelöscht} im Kontext von {Aufgaben Informationen} = parallel verfügbar • Komplexitätsgrade • Alternative: {Information}
Informationsflüsse	<ul style="list-style-type: none"> • Quelle: {Entität} (Eingangsinformation) • Ziel: {Entität} (Ausgangsinformation) • Mengengerüste
Anforderungen	<ul style="list-style-type: none"> • Zeitabhängigkeit / Verfügbarkeit • Aktualität • Genauigkeit / Vollständigkeit • Änderungshäufigkeit

Abb. 2 Modell der Aufgabenmerkmale und Merkmale bezüglich Information

Literatur

- Beck, A. ; Müller-Haffner, E. (1989): Aufgabenanalyse. Ein Überblick über Methoden zur Analyse von Arbeitsaufgaben, Projektbericht, Stuttgart, 88 S. (wird momentan aktualisiert)
- Beck, A. ; Ilg, R. (1991): Aufgabenorientierte Analyse und Gestaltung mit TASK. Erscheint in: Frese, M. ; Kasten, C. ; Zang-Scheucher, B. (Hrsg.): Software für die Arbeit von Morgen: Bilanz und Perspektiven anwendungsorientierter Forschung, Heidelberg: Springer Verlag
- Berr, M.-A. ; Feuerstein, G. ; Rödiger, K.-H. (1989): Defizite der VDI-Richtlinie "Methoden zur Analyse und Gestaltung von Arbeitssystemen im Büro" aus arbeitsorientierter Sicht. In: Maaß, S. ; Oberquelle, H. (Hrsg.): Software Ergonomie '89 Aufgabenorientierte Systemgestaltung und Funktionalität, Stuttgart : Teubner, 387-397
- Bonsiepen, L. ; Coy, W. (1990): Szenen einer Krise - Ist Knowledge Engineering eine Antwort auf die Dauerkrise des Software Engineering? In: KI, 2, 5-11
- Boose, J.H. (1989): A survey of knowledge acquisition techniques and tools. In: Knowledge Acquisition, London: Academic Press, 1, 1, 3-37
- Card, S. ; Moran, T. ; Newell, A. (1983): The Psychology of Human Computer Interaction, Hillsdale: Lawrence Erlbaum Associates
- Chen, P. (1976): The Entity-Relationship Model - Towards a Unified View of Data. In: ACM Transactions on Database Systems, 1/1, 9-36
- DeMarco, T. (1978): Structured Analysis and System Specification, Yourdon Inc., New York
- Dunckel, H. (1989): Arbeitspsychologische Kriterien zur Beurteilung und Gestaltung von Arbeitsaufgaben im Zusammenhang mit EDV-Systemen. In: Maaß, S. ; Oberquelle, H. (Hrsg.): Software-Ergonomie '89, Teubner, Stuttgart, 69-79
- Freksa, C. (1989): Wissensdarstellung und Kognitionsforschung. In: Informationstechnik, 31/2, 134-140
- Frei, F. (1981): Psychologische Arbeitsanalyse - Eine Einführung zum Thema. In: Frei, F. ; Ulich, E. (Hrsg., 1981) Beiträge zur psychologischen Arbeitsanalyse, Bern, 11-36
- Green, T. ; Schiele, F. ; Payne, S. (1988): Formalisable models of user knowledge in human-computer interaction. In: Green, T. ; Hoc, J. ; Muray, D. ; van der Veer, G. (Eds.): Theory and outcomes in human-computer interaction, London: Academic Press
- Mumford, E. ; Welter, H. (1984): Benutzerbeteiligung bei der Entwicklung von Computersystemen: Verfahren zur Steigerung der Akzeptanz und Effizienz des EDV-Einsatzes, Berlin : Schmidt Verlag
- Partridge, D. (1989): KI und das Software Engineering der Zukunft, MacGraw-Hill: Hamburg
- Polson, P. (1987): A quantitative model of human-computer interaction. In: Carroll, J. (Ed.): Interfacing Thought: Cognitive Aspects of Human-Computer Interaction, Cambridge: MIT Press, 184-235
- Rock-Evans, R. (1989): Ovum. CASE Analyst Workbenches: a Detailed Product Evaluation, Vol. 1, Ovum Ltd, London
- Rödiger, K.-H. (1987): Das Arbeitsanalyseverfahren VERA/B in der Softwareentwicklung. In: Nullmeier, E./Rödiger, K.-H. (Hrsg.): Dialogsysteme in der Arbeitswelt, Mannheim
- Rudolph, E. ; Schönfelder, E. ; Hacker, W. (1987): Tätigkeitsbewertungssystem - Geistige Arbeit, Psychodiagnostisches Zentrum Sektion Psychologie der Humboldt-Universität zu Berlin
- Schönecker, H. ; Nippa, M. (Hrsg., 1987): Neue Methoden zur Gestaltung der Büroarbeit, Baden-Baden
- Schubert, H. ; Zink, K. (1990): Partizipation - Psychologische Grundlagen eines Leitprinzips von Arbeits- und Organisationsgestaltungsmaßnahmen. In: Zeitschrift für Arbeitswissenschaft, 44, 2, 82-88
- VDI-Richtlinie 5003 (1987): Bürokommunikation, Methoden zur Analyse und Gestaltung von Arbeitssystemen im Büro, Entwurf, Verein Deutscher Ingenieure
- Volpert, W. ; Oesterreich, R. ; Gablenz-Kolakovic, S. ; Krogoll, T. ; Resch, M. (1983): Verfahren zur Ermittlung von Regulationserfordernissen in der Arbeitstätigkeit (VERA), Köln
- Ziegler, J. (1988): Aufgabenanalyse und Funktionsentwurf. In: Balzert, H. ; Hoppe, H. ; Oppermann, R. ; Peschke, H. ; Rohr, G. ; Streitz, N. (Hrsg.): Einführung in die Software-Ergonomie, Berlin: De Gruyter, 231-252